Dimensionality Reduction

- Introduction
- Subset Selection
- Feature Extraction
- Principal Component Analysis
- Linear Discriminant Analysis
- Multidimensional Scaling
- Laplacian Eigenmaps
- Locally Linear Embedding
- Isomap
- Locality Preserving Projection

Introduction (1)

- Problem of dimensionality
  - The more the number of features, the better of the performance?

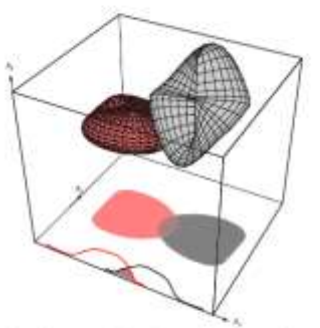


FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional x_1 subspace—there can be greater overlap of the projected distributions, and hence greater Bayes error. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

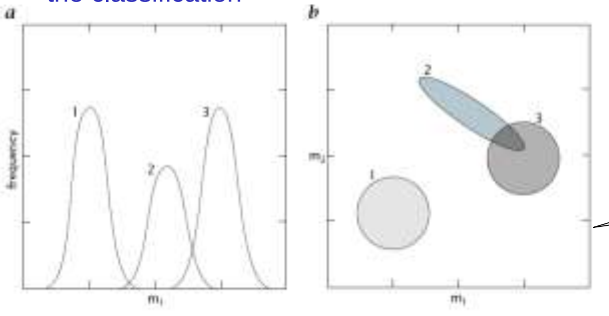
Increasing dimensionality beyond a certain point & finite number of training samples
⇒ lower performance

Possible reasons
1) Insufficient training samples
2) Using the wrong model
(e.g. the Gaussian assumption is incorrect)

Fig. 3.3 [Duda 01]

Introduction (2)

- Example
  - Increasing the number of features does not necessarily improve the classification



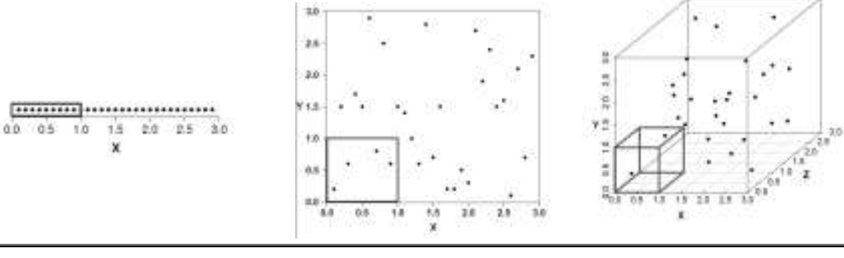
-The two features are strongly correlated
-The second feature does not tell us much new

Figure 20.5: **a** One-dimensional feature space with three object classes. **b** Extension of the feature space with a second feature. The gray shaded areas indicate the regions in which the probability for a certain class is larger than zero. The same object classes are shown in **a** and **b**.

Fig. 20.5 [B. Jahne 02]

Introduction (3)

- Problem of insufficient data
  - Demand for data increases exponentially with the dimension
    - N: the number of data needed for accurate estimates of a 1-D pdf
      - $\Rightarrow N^l$ data points would be required for an l –D space
  - Example
    - The number of instance remains the same
    - The density within the unit hypercube ↓ as the dimensions ↑



Introduction (4)

- Example
  - To grow a hypercube to contain a fraction q of the data points in a 10-D space
    - The edge length of the cube
      - $s(q) = q^{\frac{1}{l}} = q^{\frac{1}{10}}$
      - $s(0.1) = (0.1)^{\frac{1}{10}} = 0.79$
      - $s(0.01) = 0.63$
    - No longer local!!

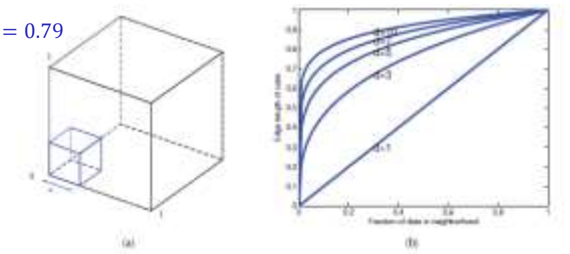


Fig. 1.16 [Murphy]

Figure 1.16 Illustration of the curse of dimensionality. (a) We embed a small cube of side s inside a larger unit cube. (b) We plot the edge length of a cube needed to cover a given volume of the unit cube as a function of the number of dimensions. Based on Figure 2.6 from (Hastie et al. 2009). Figure generated by curseDimensionality.

Introduction (5)

- Problem of insufficient data (cont.)
  - Overfitting
    - When fitting highly flexible models
      - Not to model every minor variation in the data
  - Handling insufficient data
    - Reduce the dimensionality
      - Dimension reduction
    - Bayesian technique
      - Assume a reasonable prior on the parameters
    - Model simplification
      - Statistical independence assumption (e.g., naïve Bayes model)
    - Heuristics
      - e.g. thresholding the estimated covariance matrix such that only correlations above a threshold are retained

Introduction (6)

- Dimension reduction
  - Reducing the number of features to a sufficient minimum
    - To avoid curse of dimensionality
      - Large number of data points are required in a high-dimensional space
    - To reduce computational complexity
  - Example: binary handwritten digit 
    - Each digit contains $28 \times 28 = 784$ pixels
      - A point in the 784 dimensional space
    - The number of possible images is $2^{784} \approx 10^{236}$!
    - We need only a few examples to understand how to recognize a digit
- Feature selection
  - In the measurement space (subset selection)
  - In the transformed space (also called feature extraction)

Introduction (7)

- Feature selection
  - Subset selection in the measurement space
  - To seek a subset l-dimensional features out of the available d features $\mathbf{x} = [x_1, \dots, x_d]^T$
    - By optimizing some criterion J over all possible subsets X_l of size l
      - $J(\tilde{X}_l) = \max_{X \in X_l} J(X)$
- Feature extraction
  - Feature selection in the transformed space
  - To extract a new set of features by transforming the original $\mathbf{x}$
    - By optimizing some J over all possible transformations
      - $J(\tilde{A}) = \max_{A \in A_l} J(A(\mathbf{x}))$

Subset Selection (1)

- To seek a subset of features
  - $J(\tilde{X}_l) = \max_{X \in X_l} J(X)$
  - $J(.)$: criteria for feature selection
    - Classifier dependent criteria
      - Choose the set for which the classifier performs well on a validation set
      - Different choices of classifier may result in different feature sets
    - Classifier independent criteria
      - Choose the feature set that achieve maximal separability of the data
  - Selection approaches
    - The simplest one: exhaustive search
      - Computationally prohibitive!!
      - Example
        - » To select a subset of $l = 12$ features out of $d = 24$ features
        - » There are about 2.7 million possible feature subsets

Subset Selection (2)

- Scalar feature selection
  - Features are treated individually
    - Features are ranked by $J(x_k), k = 1, \dots, d$ computed for each single feature x_k
    - Selecting the l features with the largest $J(x_k)$ values
- Feature vector selection
  - Selecting the best feature vector combination l features
    - The total number of combinations are $\binom{d}{l} = \frac{d!}{l!(d-l)!}$
      - If $d = 20, l = 5$, there are 15504 possible feature subsets
  - Heuristics have been used to avoid exhaustive search
    - Sequential methods
      - Do not examine all possible subsets
      - No guarantee of finding the optimal subset

Subset Selection (3)

- Sequential forward selection (*Bottom-up method*)
  - To build up a set F of l features incrementally
    - Starting with the empty set $F = \emptyset$
    - At each step, for all possible x_k, calculate the criterion and choose the one that maximize the criterion
      - $j = \underset{k}{\operatorname{argmax}} J(F \cup \{x_k\})$
    - Add the selected feature x_j to F if
      - $J(F \cup \{x_j\}) > J(F)$
    - Stop if adding any feature does not increase J
  - This is a greedy procedure and does not guarantee finding the optimal subset
  - Nesting problem
    - Once a feature is chosen, it cannot be discarded

Subset Selection (4)

- Example (p.117, [Alpaydin, 2014])
  - 3 classes
    - 50 instances per class: 20 for training, 30 for validation
  - Feature dimension
    - $d = 4$
  - Minimum distance classifier
  - Step 1
    - When using single features separately
      - Validation accuracies
        - » $J(\{x_1\}) = 0.76$
        - » $J(\{x_2\}) = 0.57$
        - » $J(\{x_3\}) = 0.92$
        - » $J(\{x_4\}) = 0.94$
      - $F = \{x_4\}$

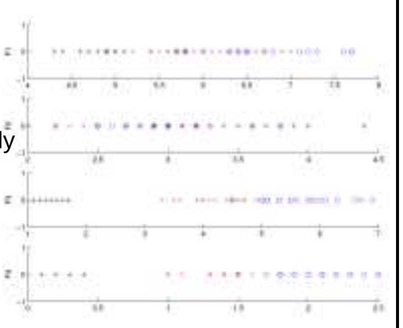


Fig. 6.1 [Alpaydin]

Subset Selection (5)

- Example (cont.)
  - Step 2
    - Validation accuracies
      - $J(F \cup \{x_1\}) = 0.87$
      - $J(F \cup \{x_2\}) = 0.92$
      - $J(F \cup \{x_3\}) = 0.96$
    - $F \leftarrow F \cup \{x_3\}$
  - Step 3
    - Validation accuracies
      - $J(F \cup \{x_1\}) = 0.94$
      - $J(F \cup \{x_2\}) = 0.94$
    - Because $J(F \cup \{x_k\}) \ngtr J(F)$
      - Let $F = \{x_3, x_4\}$

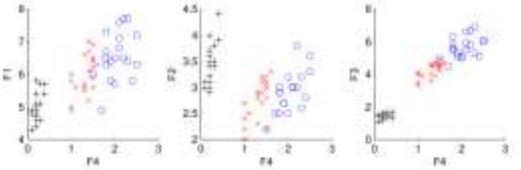


Fig. 6.2 [Alpaydin]

Subset Selection (6)

- Floating search method
  - A modification on sequential forward selection
  - At each step
    - Inclusion of a new feature that increases J the greatest
      - $F_{k+1} \leftarrow F_k \cup \{x_{k+1}\}$
    - Finding the feature that reduces J the least when it is removed from F_{k+1}
      - » $x_r = \underset{y \in F_{k+1}}{\operatorname{argmax}} J(F_{k+1} - \{y\})$ Conditional exclusion
      - » If $x_r = x_{k+1}$
        - » $k \leftarrow k + 1$, go back to inclusion step
      - » Otherwise, remove x_r from F_{k+1}
      - » $F'_k \leftarrow F_{k+1} - \{x_r\}$
    - Continue removing features from F'_k while
      - $J(F'_{k-1}) > J(F'_k)$

Subset Selection (7)

- Sequential backward selection (*Top-down method*)
  - To start with the full set of d features and remove redundant features successively
    - Starting with the full feature set $F = \{x_1, \dots, x_d\}$
    - At each step, find the one that maximize the criterion
      - $j = \underset{k}{\operatorname{argmax}} J(F - x_k)$
    - Remove x_j from F if
      - $J(F - x_j) > J(F)$
    - Stop if removing any feature does not increase J
  - Nesting problem
    - Once a feature is discarded, there is no possibility for it to be reconsidered again

Feature Extraction

- Feature selection in the transformed space
  - The optimization is performed over all possible transformations of the high-dimensional features $\mathbf{x} \in R^d$
    - $J(\tilde{A}) = \max_{A \in A_l} J(A(\mathbf{x}))$
  - The transformed feature vector
    - $\mathbf{z} \in J\left(\tilde{A}(\mathbf{x})\right) \in R^l$
  - The transform is chosen so that the transformed features
    - Pack most of the classification information in a small number of features, or
    - Have optimized class separability criterion, or
    - Have the minimal reconstruction error, or
    - Become mutually independent
    - …

Principal Component Analysis (1)

- Principal component analysis (PCA)
  - Also known as Karhunen-Loeve Transform (KLT)
  - The transformation matrix is computed in an unsupervised mode
  - Given the data $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \boldsymbol{x}_i \in R^d$
    - Assume zero-mean, for notational simplicity
      - $\boldsymbol{\mu}_\mathbf{x} = \mathbf{0}$
  - To project the data to a lower dimensional linear space such that
    - The average reconstruction error is minimized
      - $\min J(\mathbf{W}, \mathbf{Z}) = \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{W}\mathbf{z}_i\|^2$
    - Or, the variance of the projected data is maximized
      - $\max \sum_{i=1}^{N} \sigma_{\mathbf{z}_i}^2$
    - The 2 definitions give rise to the SAME algorithm

Principal Component Analysis (2)

- Maximum variance formulation
  - Given the data $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \mathbf{x}_i \in R^d$
  - To project the data onto a l-dimensional space ($l < d$) while maximizing the variance of the projected data
    - $\max \sum_{i=1}^{N} \sigma_{\mathbf{z}_i}^2$
      - $\mathbf{z}_i = \mathbf{W}^T(\mathbf{x}_i - \boldsymbol{\mu}_\mathbf{x}) \in R^l$
      - $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_l] \in R^{d \times l}$

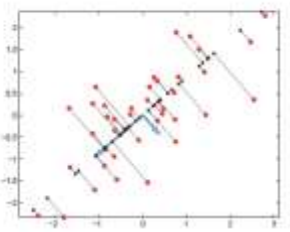


Fig. 15.2 [Barber 14]

Figure 15.2: Projection of two dimensional data using one dimensional PCA. Plotted are the original datapoints $\mathbf{x}$ (larger rings) and their reconstructions $\tilde{\mathbf{x}}$ (small dots) using 1 dimensional PCA. The lines represent the orthogonal projection of the original datapoint onto the first eigenvector. The arrows are the two eigenvectors scaled by the square root of their corresponding eigenvalues. The data has been centred to have zero mean. For each 'high dimensional' datapoint $\mathbf{x}$, the 'low dimensional' representation $\mathbf{y}$ is given in this case by the distance (possibly negative) from the origin along the first eigenvector direction to the corresponding orthogonal projection point.

Principal Component Analysis (3)

- Maximum variance formulation (cont.)
  - First search for the single direction $\mathbf{w}_1 \in R^d$, $\|\mathbf{w}_1\| = 1$
    - Each data point $\mathbf{x}_i$ is then projected to a scalar value $z_{1,i} = \mathbf{w}_1^T \mathbf{x}_i$
    - Then the variance of the projected data
      - $\sum_{i=1}^{N} \sigma_{z_{1,i}}^2 = \frac{1}{N}\sum_{i=1}^{N}\left(z_{1,i} - \mu_{z_1}\right)^2 = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{w}_1^T\mathbf{x}_i - \mathbf{w}_1^T\boldsymbol{\mu}_\mathbf{x})^2$

        $= \mathbf{w}_1^T\left[\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu}_\mathbf{x})(\mathbf{x}_i - \boldsymbol{\mu}_\mathbf{x})^T\right]\mathbf{w}_1$

        $= \mathbf{w}_1^T \boldsymbol{\Sigma}_\mathbf{x} \mathbf{w}_1$
    - To maximize $\mathbf{w}_1^T \boldsymbol{\Sigma}_\mathbf{x} \mathbf{w}_1$ subject to $\|\mathbf{w}_1\|^2 = \mathbf{w}_1^T\mathbf{w}_1 = 1$
      - $L = \mathbf{w}_1^T \boldsymbol{\Sigma}_\mathbf{x} \mathbf{w}_1 - \alpha(\mathbf{w}_1^T\mathbf{w}_1 - 1)$
      - Let $\frac{\partial L}{\partial \mathbf{w}_1} = 2\boldsymbol{\Sigma}_\mathbf{x}\mathbf{w}_1 - 2\alpha\mathbf{w}_1 = 0$

        $\frac{\partial}{\partial \mathbf{x}}[\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x}] = [\boldsymbol{M} + \boldsymbol{M}^T]\boldsymbol{x}$
      - $\mathbf{w}_1$ is the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{\Sigma}_\mathbf{x}$
    - Thus
      - $\sum_{i=1}^{N} \sigma_{z_{i,1}}^2 = \mathbf{w}_1^T \boldsymbol{\Sigma}_\mathbf{x} \mathbf{w}_1 = \lambda_1 = \alpha$

Principal Component Analysis (4)

- Maximum variance formulation (cont.)
  - The next optimal direction $\mathbf{w}_2$ should
    - Maximize variance, be of unit length $\|\mathbf{w}_2\| = 1$, be orthogonal to $\mathbf{w}_1$
    - That is, to maximize $\mathbf{w}_2^T \boldsymbol{\Sigma}_\mathbf{x} \mathbf{w}_2$ subject to $\|\mathbf{w}_2\|^2 = 1$ and $\mathbf{w}_2^T\mathbf{w}_1 = 0$
      - $L = \mathbf{w}_2^T \boldsymbol{\Sigma}_\mathbf{x} \mathbf{w}_2 - \alpha(\mathbf{w}_2^T\mathbf{w}_2 - 1) - \beta(\mathbf{w}_2^T\mathbf{w}_1 - 0)$
      - Let $\frac{\partial L}{\partial \mathbf{w}_2} = 2\boldsymbol{\Sigma}_\mathbf{x}\mathbf{w}_2 - 2\alpha\mathbf{w}_2 - \beta\mathbf{w}_1 = 0$

        $\Rightarrow 2\mathbf{w}_1^T\boldsymbol{\Sigma}_\mathbf{x}\mathbf{w}_2 - 2\alpha\mathbf{w}_1^T\mathbf{w}_2 - \beta\mathbf{w}_1^T\mathbf{w}_1 = 0 \Rightarrow 2\mathbf{w}_1^T\boldsymbol{\Sigma}_\mathbf{x}\mathbf{w}_2 - \beta = 0$

        $\Rightarrow \mathbf{w}_1^T\boldsymbol{\Sigma}_\mathbf{x}\mathbf{w}_2 = (\boldsymbol{\Sigma}_\mathbf{x}\mathbf{w}_1)^T\mathbf{w}_2 = \lambda_1\mathbf{w}_1^T\mathbf{w}_2 = 0 \Rightarrow \beta = 0$

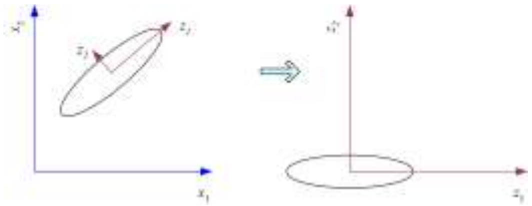
        $\Rightarrow \boldsymbol{\Sigma}_\mathbf{x}\mathbf{w}_2 = \alpha\mathbf{w}_2$
      - $\mathbf{w}_2$ is the eigenvector of $\boldsymbol{\Sigma}_\mathbf{x}$ with the 2nd largest eigenvalue $\lambda_2 = \alpha$
  - We can incrementally choose each new direction $\mathbf{w}_j$
    - Orthogonal to those already considered
    - The sum of the variances of the l principal components = sum of eigenvalues $\sum_{i=1}^{N}\sigma_{\mathbf{z}_i}^2 = \sum_{i=1}^{l}\lambda_i$

Principal Component Analysis (5)

- To summarize, PCA involves
  - Evaluating the covariance matrix $\mathbf{\Sigma_x}$
  - Finding the l eigenvectors $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_l\}$ of $\mathbf{\Sigma_x}$ corresponding to the l largest eigenvalues $\mathbf{\Sigma_x}$
    - If $\mathbf{\Sigma_x}$ is positive definite (i.e., $\mathbf{x}^T\mathbf{\Sigma_x}\mathbf{x} > 0, \forall \mathbf{x} \neq \mathbf{0}$)
      - All its eigenvalues are positive
    - If $\mathbf{\Sigma_x}$ is singular
      - Its rank k is smaller than d, i.e. $k < d$ and $\lambda_i = 0, i = k+1, \dots, d$
      - The k eigenvectors with nonzero eigenvalues are the dimensions of the reduced space
  - The transformation
    - $\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})$
      - $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_l]$
      - $\boldsymbol{\mu}_\mathbf{z} = \mathbf{0}$

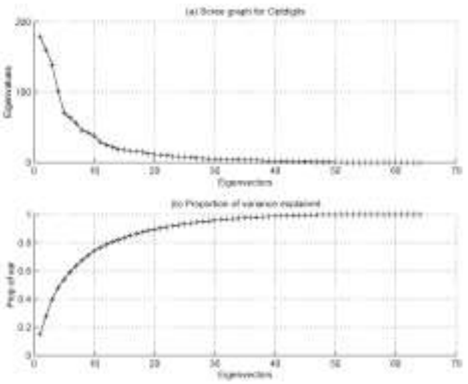
Principal Component Analysis (6)

- Total variance
  - $\sum_{i=1}^{N} \sigma_{z_i}^2 = \sum_{i=1}^{d} \lambda_i \quad (\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d)$
- Proportion of variance
  - The first l principal components account for
    - $\frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$ of the total variance
  - Given a specified percentage P, we can choose l so that
    - $\sum_{i=1}^{l} \lambda_i \geq P \sum_{i=1}^{d} \lambda_i \geq \sum_{i=1}^{l-1} \lambda_i$
    - If the dimensions are highly correlated
      - There will be a small number of eigenvectors with large eigenvalues
    - Otherwise
      - There will be no much gain through PCA



PCA centers the sample and rotates the axes to line up with the directions of higher variance

Fig. 6.3 [Alpaydin]



A handwritten digit dataset
- 10 classes
- 64-dimensional input

The first 20 eigenvectors explain 90% of the variance

Fig. 6.4 [Alpaydin]

Principal Component Analysis (7)

- Another derivation
  - Let $\mathbf{X}$ be the centered data matrix
    - $\mathbf{X} = [\mathbf{x}_1 - \boldsymbol{\mu}_\mathbf{x}, \mathbf{x}_2 - \boldsymbol{\mu}_\mathbf{x}, \dots, \mathbf{x}_N - \boldsymbol{\mu}_\mathbf{x}] \in R^{d\times N}$
  - The covariance matrix
    - $\boldsymbol{\Sigma}_\mathbf{x} = \mathbf{X}\mathbf{X}^T \in R^{d\times d}$
  - To decorrelate the original dimension by finding $\mathbf{W}$ such that
    - $\boldsymbol{\Sigma}_\mathbf{z} =$ diagonal matrix
      - $\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})$, $\mathbf{Z} = \mathbf{W}^T\mathbf{X}$
  - If we form a $d \times d$ matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_d]$
    - $\mathbf{w}_i$ is the normalized eigenvector of $\boldsymbol{\Sigma}_\mathbf{x}$
    - $\mathbf{W}^T\mathbf{W} = \mathbf{I}$
  - Then the symmetric matrix $\boldsymbol{\Sigma}_\mathbf{x}$ can be diagonalized by
    - $\mathbf{W}^T\boldsymbol{\Sigma}_\mathbf{x}\mathbf{W} = \boldsymbol{\Lambda}$
    - $\boldsymbol{\Sigma}_\mathbf{z} = \mathbf{Z}\mathbf{Z}^T = \mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W} = \mathbf{W}^T\boldsymbol{\Sigma}_\mathbf{x}\mathbf{W} = \boldsymbol{\Lambda}$

Principal Component Analysis (8)

- Example 6.3 (p. 334 [Theodoridis et. al.])
  - 100 points are generated by the model
    - $x_2 = x_1 + \varepsilon$, where ε is uniformly distributed in $[-0.5,0.5]$
  - After performing an eigendecomposition on the covariance matrix
    - The resulting eigenvectors and eigenvalues
      - $\mathbf{a}_0 = \mathbf{w}_1 = \begin{bmatrix} 0.7045 \\ 0.7097 \end{bmatrix}$
      - $\mathbf{a}_1 = \mathbf{w}_2 = \begin{bmatrix} -0.7097 \\ 0.7045 \end{bmatrix}$
      - $\lambda_1 = 17.26$
      - $\lambda_2 = 0.04$
      - $\lambda_1 \gg \lambda_2$

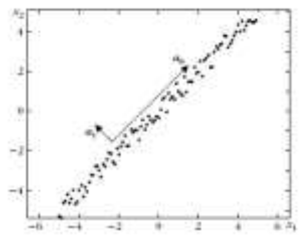


FIGURE 6.2
Points around the $x_2 = x_1$ line. The eigenvectors of the associated covariance matrix are a_0 and a_1. The principal eigenvector a_0 points in the direction of maximum variance.

Principal Component Analysis (9)

- PCA
  - Is sensitive to outliers
  - Does not necessary lead to maximum class separability in the lower dimensional space
    - Because PCA seeks a projection that best represents the data in the least square sense
    - But not optimized w.r.t. class separability

The sample is most spread out along $\mathbf{a}_1$

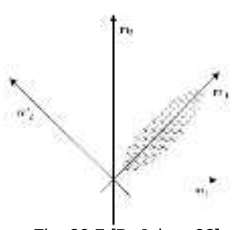


Fig. 20.7 [B. Jahne 02]

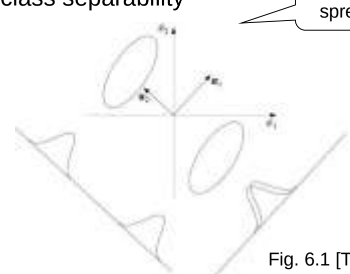


Fig. 6.1 [Theodoridis et. al.]

Principal Component Analysis (10)

- Example (dimension reduction)
  - The digit data set
    - Representing the eigenvectors as images of the same size as the data (dimension $d = 28 \times 28 = 783$)

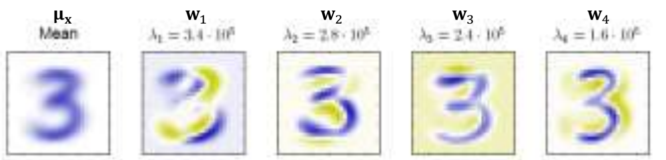


Fig. 12.3 [Bishop,06]

    - The PCA reconstruction $\hat{\mathbf{x}} = \boldsymbol{\mu}_{\mathbf{x}} + \sum_{i=1}^{l} z_i \mathbf{w}_i$

Fig. 12.5 [Bishop,06]

Principal Component Analysis (11)

- Example (data pre-processing)
  - Whitening
    - $\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})$
      - $\boldsymbol{\Sigma}_{\mathbf{z}} = \mathbf{W}^T \boldsymbol{\Sigma}_{\mathbf{x}} \mathbf{W} = \boldsymbol{\Lambda}$
    - Let $\mathbf{W}_{\mathbf{w}} = \mathbf{W}\boldsymbol{\Lambda}^{-\frac{1}{2}}$, $\mathbf{z}_{\mathbf{w}} = \mathbf{W}_{\mathbf{w}}^T(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})$
      - $\boldsymbol{\Sigma}_{\mathbf{z}_{\mathbf{w}}} = \mathbf{W}_w^T \boldsymbol{\Sigma}_{\mathbf{x}} \mathbf{W}_{\mathbf{w}} = \boldsymbol{\Lambda}^{-\frac{1}{2}} \mathbf{W}^T \boldsymbol{\Sigma}_{\mathbf{x}} \mathbf{W} \boldsymbol{\Lambda}^{-\frac{1}{2}} = \mathbf{I}$

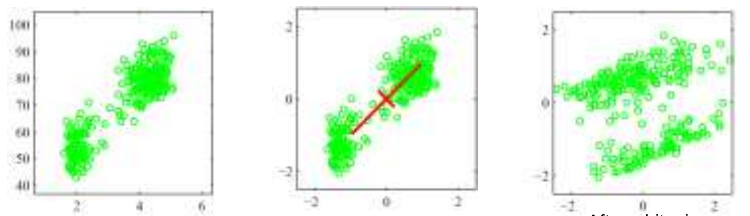


Fig. 12.6 [Bishop,06]

The original data

Normalized data (zero mean, unit variance)

After whitening (zero mean, unit variance)

Eigenfaces (1)

- Face recognition using Eigenfaces [Turk and Pentland, 1991]
  - Training data matrix
    - $\mathbf{X} = [\mathbf{x}_1 - \boldsymbol{\mu}_\mathbf{x}, \mathbf{x}_2 - \boldsymbol{\mu}_\mathbf{x}, \dots, \mathbf{x}_N - \boldsymbol{\mu}_\mathbf{x}] \in R^{d \times N}, N \ll d$ (e.g., $d = 256^2$)
  - To determine the l eigenvalues and eigenvectors in a large matrix $\boldsymbol{\Sigma}_\mathbf{x} = \mathbf{X}\mathbf{X}^T \in R^{d \times d}$ is intractable
    - Consider the eigenvectors of $\mathbf{X}^T\mathbf{X} \in R^{N \times N}$
      - $\mathbf{X}^T\mathbf{X}\mathbf{b}_i = \lambda_i \mathbf{b}_i$
      - $\mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{b}_i = \lambda_i \mathbf{X}\mathbf{b}_i$
      - $\mathbf{X}\mathbf{X}^T\mathbf{w}_i = \lambda_i \mathbf{w}_i$
        - » $\mathbf{X}\mathbf{b}_i$ are the eigenvectors of $\boldsymbol{\Sigma}_\mathbf{x}$
    - 1) construct the $N \times N$ matrix $\mathbf{X}^T\mathbf{X}$
    - 2) find the l $(l < N)$ eigenvectors $\mathbf{b}_i$
    - 3) form the eigenfaces $\mathbf{w}_i = \mathbf{X}\mathbf{b}_i, i = 1, \dots, l$

mean face

Eigenfaces (2)

- Training
  - For the kth individual, calculate the averaged vector
    - $\mathbf{z}_k = \frac{1}{|subject_k|} \sum_{\mathbf{x}_i \in subject_k} \mathbf{W}^T(\mathbf{x}_i - \boldsymbol{\mu}_\mathbf{x})$
- Recognition
  - For each new face, calculate
    - $\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})$
  - Measure the distance to each class
    - $\varepsilon_k = \|\mathbf{z} - \mathbf{z}_k\|^2$
  - Classify the input to an individual if
    - The minimum distance $< T1$, and
    - The reconstructed error $< T$

Linear Discriminant Analysis (1)

- Fisher linear discriminant analysis (FLDA)
  - The class labels are assumed known
  - To transform to a space of dimension at most $K-1$ (K: #classes)
    - No assumption is made regarding to the data distribution
    - The axes of the transformed coordinate system can be ordered in terms of importance for discrimination
  - Consider optimizing the scatter matrices criterion J on the transformed $\mathbf{z}$
    - Given $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}, \boldsymbol{x}_i \in R^d$ in a classification task of K classes
    - Compute
      - $\mathbf{z}_\mathrm{i} = \mathbf{W}^T\mathbf{x}_\mathrm{i} \in R^l; \quad \mathbf{W} \in R^{d \times l}$
    - The criterion
      - $J_3 = trace\left\{\widetilde{\mathbf{S}_W^{-1}}\widetilde{\mathbf{S}_B}\right\}$

Scatter Matrices (1)

- Within-class scatter matrix
  - $\mathbf{S}_W = \sum_{i=1}^K P_i \mathbf{S}_i = \frac{1}{N}\sum_{i=1}^K \sum_{\mathbf{x}\in C_i}(\mathbf{x}-\boldsymbol{\mu}_i)(\mathbf{x}-\boldsymbol{\mu}_i)^T$
    - where $\mathbf{S}_i$ is the sample covariance matrix for class C_i
      - $\mathbf{S}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in C_i}(\mathbf{x}-\boldsymbol{\mu}_i)(\mathbf{x}-\boldsymbol{\mu}_i)^T$
      - $\boldsymbol{\mu}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in C_i}\mathbf{x}$
      - $P_i = \frac{n_i}{N}$
    - Is symmetric and positive semidefinite
    - Is usually nonsingular if $N > d$
  - Trace($\mathbf{S}_W$) is a measure of the averaged variance (over all classes)
    - $tr(\mathbf{S}_W) = \sum_{i=1}^K P_i tr(\mathbf{S}_i) = \frac{1}{N}\sum_{i=1}^K \sum_{\mathbf{x}\in C_i}\|\mathbf{x}-\boldsymbol{\mu}_i\|^2$

Scatter Matrices (2)

- Between-class scatter matrix
  - $\mathbf{S}_B = \sum_{i=1}^{K} P_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$
    - where $\boldsymbol{\mu}$ is the global mean vector
      - $\boldsymbol{\mu} = \frac{1}{N}\sum \mathbf{x} = \sum_{i=1}^{K} P_i \boldsymbol{\mu}_i$
    - Is symmetric and positive semidefinite
    - Its rank is at most $K-1$
      - Because it is the sum of K $(d \times d)$ matrices of rank one or less
        - » Only $K-1$ of these are independent
  - Trace($\mathbf{S}_B$) is a measure of the averaged distance of $\boldsymbol{\mu}_i$ from $\boldsymbol{\mu}$
    - $tr(\mathbf{S}_B) = \sum_{i=1}^{K} P_i \|\boldsymbol{\mu}_i - \boldsymbol{\mu}\|^2$

Scatter Matrices (3)

- Mixture scatter matrix (or total scatter matrix)
  - $\mathbf{S}_M = \frac{1}{N}\sum_{\mathbf{x}}(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T$

$$= \frac{1}{N}\sum_{i=1}^{K}\sum_{\mathbf{x}\in C_i}(\mathbf{x} - \boldsymbol{\mu}_i + \boldsymbol{\mu}_i - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu}_i + \boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$
$$= \frac{1}{N}\sum_{i=1}^{K}\sum_{\mathbf{x}\in C_i}(\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T + \frac{1}{N}\sum_{i=1}^{K}\sum_{\mathbf{x}\in C_i}(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$
$$= \mathbf{S}_W + \sum_{i=1}^{K} P_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$
$$= \mathbf{S}_W + \mathbf{S}_B$$

  - Is the sample covariance matrix of the feature vector
  - Trace($\mathbf{S}_M$) is the sum of variances around the global mean
    - Is independent of how the samples are partitioned
    - $tr(\mathbf{S}_M) = tr(\mathbf{S}_W) + tr(\mathbf{S}_B) = \frac{1}{N}\sum_{\mathbf{x}}\|\mathbf{x} - \boldsymbol{\mu}\|^2$

Scatter Matrices (4)

- Criterion based on scatter matrices
  - A class separability measure should have larger values when the within-class spread is small and the between-class spread is large
    - $J_1 = \frac{tr(\mathbf{S}_M)}{tr(\mathbf{S}_W)}$ or $\frac{tr(\mathbf{S}_B)}{tr(\mathbf{S}_W)}$
    - $J_2 = \frac{|\mathbf{S}_M|}{|\mathbf{S}_W|} = |\mathbf{S}_W^{-1}\mathbf{S}_M|$
    - $J_3 = tr(\mathbf{S}_W^{-1}\mathbf{S}_M)$ or $tr(\mathbf{S}_W^{-1}\mathbf{S}_B)$

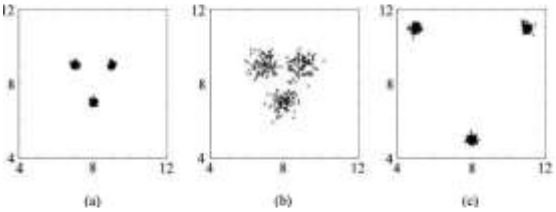


Fig. 5.5 [Theodoridis et. al.]

$J_3 = 164.7$ $J_3 = 12.5$ $J_3 = 620.9$

Scatter Matrices (5)

- The scatter matrices after linear transformation
  - $\widetilde{\mathbf{S}_W} = \frac{1}{N}\sum_{i=1}^{K}\sum_{\mathbf{z}\in C_i}(\mathbf{z}-\widetilde{\boldsymbol{\mu}_i})(\mathbf{z}-\widetilde{\boldsymbol{\mu}_i})^T$
    $= \frac{1}{N}\sum_{i=1}^{K}\sum_{\mathbf{x}\in C_i}(\mathbf{W}^T\mathbf{x}-\mathbf{W}^T\boldsymbol{\mu}_i)(\mathbf{W}^T\mathbf{x}-\mathbf{W}^T\boldsymbol{\mu}_i)^T$
    $= \mathbf{W}^T\mathbf{S}_W\mathbf{W}$
  - $\widetilde{\mathbf{S}_B} = \sum_{i=1}^{K}P_i(\widetilde{\boldsymbol{\mu}_i}-\widetilde{\boldsymbol{\mu}})(\widetilde{\boldsymbol{\mu}_i}-\widetilde{\boldsymbol{\mu}})^T$
    $= \sum_{i=1}^{K}P_i(\mathbf{W}^T\boldsymbol{\mu}_i-\mathbf{W}^T\boldsymbol{\mu})(\mathbf{W}^T\boldsymbol{\mu}_i-\mathbf{W}^T\boldsymbol{\mu})^T$
    $= \mathbf{W}^T\mathbf{S}_B\mathbf{W}$
- The criterion
  - $J_3(\mathbf{W}) = trace\left\{\widetilde{\mathbf{S}_W}^{-1}\widetilde{\mathbf{S}_B}\right\} = trace\{(\mathbf{W}^T\mathbf{S}_W\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{S}_B\mathbf{W})\}$

Linear Discriminant Analysis (2)

- Two-category case ($K = 2, l = 1$)
  - The scatter matrices
    - $\mathbf{S}_W = P_1\mathbf{S}_1 + P_2\mathbf{S}_2 = \frac{1}{N}\sum_{i=1}^{2}\sum_{\mathbf{x}\in C_i}(\mathbf{x}-\boldsymbol{\mu}_i)(\mathbf{x}-\boldsymbol{\mu}_i)^T$
    - $\mathbf{S}_B = \sum_{i=1}^{2} P_i(\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^T = P_1P_2(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T$
      - The between-class matrix is proportional to the outer product of 2 vectors
      - Its rank is at most one
  - Goal
    - To find an orientation for which the projected data are well separated
    - $z = \mathbf{w}^T\mathbf{x};\ \mathbf{w} \in R^{d\times 1}$
      - The magnitude of $\mathbf{w}$ is of no significance
      - The direction of $\mathbf{w}$ is important

Linear Discriminant Analysis (3)

- Two-category case (cont.)
  - $z = \mathbf{w}^T\mathbf{x};\ \mathbf{w} \in R^{d\times 1}$
    - $J(\mathbf{w}) = \frac{\widetilde{S_B}}{\widetilde{S_W}} = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$
    - This generalized Rayleigh quotient is maximized if $\mathbf{w}$ is chosen such that $\boldsymbol{S}_B\mathbf{w} = \lambda\boldsymbol{S}_W\mathbf{w}$, where λ is the largest eigenvalue of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$

- Rayleigh quotient
  - $R(\mathbf{w}) = \frac{\mathbf{w}^T S \mathbf{w}}{\mathbf{w}^T\mathbf{w}}$
  - If $\boldsymbol{S} = \boldsymbol{S}^T$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ and orthnormal eigenvectors $\boldsymbol{u}_i$
  - Then $\forall \mathbf{w} \neq \mathbf{0}, \mathbf{w} \in R^{d\times 1}$
    - $\lambda_d \leq R(\mathbf{w}) \leq \lambda_1$
  - $R(\mathbf{w}) = \lambda_1$
    - When $\mathbf{w}$ is chosen as an eigenvector of $\boldsymbol{S}$ corresponding to λ_1

Linear Discriminant Analysis (4)

- Generalized Rayleigh quotient
  - $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$
    - Let $\mathbf{b} = \mathbf{S}_W^{\frac{1}{2}} \mathbf{w} \Longrightarrow \mathbf{w} = \mathbf{S}_W^{-\frac{1}{2}} \mathbf{b}$
    - $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \frac{\mathbf{w}^T \mathbf{S}_W^{-1/2} \mathbf{S}_B \mathbf{S}_W^{-1/2} \mathbf{w}}{\mathbf{b}^T \mathbf{b}}$ is maximized when $\mathbf{b}$ is chosen such that
      - $\mathbf{S}_W^{-1/2} \mathbf{S}_B \mathbf{S}_W^{-1/2} \mathbf{b} = \lambda \mathbf{b}$
        $\Longrightarrow \mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{w} = \lambda \mathbf{w}$
        $\Longrightarrow \mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}$
        - λ is the largest eigenvalue of $\mathbf{S}_W^{-1} \mathbf{S}_B$
    - The corresponding maximum value is
      - $\frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \frac{\lambda \mathbf{w}^T \mathbf{S}_W \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \lambda$

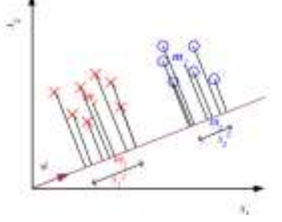
Linear Discriminant Analysis (5)

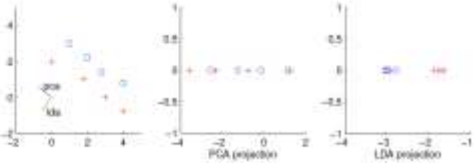
- Two-category case (cont.)
  - Or, from
    - $J(\mathbf{w}) = \frac{\widetilde{S_B}}{\widetilde{S_W}} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$
    - J is invariant w.r.t. rescaling of the vector $\mathbf{w} \to \alpha \mathbf{w}$
      - We can choose $\mathbf{w}$ such that the denominator is simply $\mathbf{w}^T \mathbf{S}_W \mathbf{w} = 1$
    - Maximizing J = the constraint optimization problem
      - $\max_{\mathbf{w}} \mathbf{w}^T \mathbf{S}_B \mathbf{w}$ subject to $\mathbf{w}^T \mathbf{S}_W \mathbf{w} = 1$
    - The Largrangian
      - $L = \mathbf{w}^T \mathbf{S}_B \mathbf{w} + \lambda (1 - \mathbf{w}^T \mathbf{S}_W \mathbf{w})$
      - $\frac{\partial L}{\partial \mathbf{w}} = 2\mathbf{S}_B \mathbf{w} - \lambda 2 \mathbf{S}_W \mathbf{w} = 0$
      - $\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}$ Generalized eigen-problem

Linear Discriminant Analysis (6)

- Two-category case (cont.)
  - In this particular case, it is unnecessary to solve for the eigenvalues and eigenvectors of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$
    - Because $\mathbf{S}_B = P_1 P_2 (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$
      - $\mathbf{S}_B \propto (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$
    - $\boldsymbol{S}_B \mathbf{w} = \lambda \boldsymbol{S}_W \mathbf{w}$
      - $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \cdot \left((\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \mathbf{w}\right) = \lambda \boldsymbol{S}_W \mathbf{w}$
        - » $\mathbf{w}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = a = scalar$
      - $a(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = \lambda \boldsymbol{S}_W \mathbf{w}$
      - $\mathbf{w} \propto \boldsymbol{S}_W^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$
    - Example ($P_1 = P_2$)
      - $\widetilde{S_W} = \frac{1}{2}(\sigma_{z,1}^2 + \sigma_{z,2}^2)$
      - $\widetilde{S_B} = \frac{1}{4}\left(\mu_{z,1} - \mu_{z,2}\right)^2$
      - $J(\mathbf{w}) = \frac{\widetilde{S_B}}{\widetilde{S_W}} \propto \frac{(\mu_{z,1} - \mu_{z,2})^2}{\sigma_{z,1}^2 + \sigma_{z,2}^2}$

Figs. 6.10-6.11 [Alpaydin]





Linear Discriminant Analysis (7)

- Two-category case (cont.)

$$
\begin{aligned}
P_1 &= P_2 \\
\mathbf{S}_1 = \mathbf{S}_2 = \mathbf{S}_W &= \boldsymbol{\Sigma} \\
\Rightarrow \mathbf{w} &\propto \boldsymbol{S}_W^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \\
&= \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)
\end{aligned}
$$

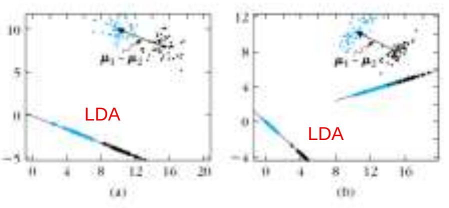


Fig. 5.6 [Theodoridis 09]

FIGURE 5.6

(a) The optimal line resulting from Fisher's criterion, for two Gaussian classes. Both classes share the same diagonal covariance matrix, with equal elements on the diagonal. The line is parallel to $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. (b) The covariance matrix for both classes is nondiagonal. The optimal line is on the left. Observe that it is no more parallel to $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. The line on the right is not optimal and the classes, after the projection, overlap.

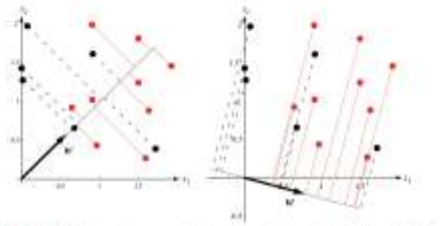


FIGURE 3.5. Projection of the same set of samples onto two different lines in the directions marked **w**. The figure on the right shows greater separation between the red and black projected points. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Fig. 3.5 [Duda 01]

Linear Discriminant Analysis (8)

- Two-category case (cont.)
  - The Fisher's linear discriminant is optimal if the classes are normally distributed with equal covariance matrices
    - Although the discrimanant direction $\mathbf{w}$ has been derived *without* any assumptions of normality $\mathbf{w} \propto \boldsymbol{S}_W^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$
  - Review (Bayesian classification for Normal distributions, case 3)
    - $g_i(\mathbf{x}) = \ln p(\mathbf{x}|C_i) + \ln P(C_i)$
      - $p(\mathbf{x}|C_i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{l}{2}}\sqrt{|\boldsymbol{\Sigma}|}} \exp(-\frac{(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}{2})$
    - $g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x}) = 0$
    - $\mathbf{w}^T(\mathbf{x} - \mathbf{x_0}) = 0$
      - $\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1)$
      - $\mathbf{x_0} = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) - \frac{1}{(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)} \ln\frac{P(C_1)}{P(C_2)}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$

Multiple Discriminant Analysis (1)

- Generalization to multiple class problem
  - The projection is from a d–dimensional space to an l-dimensional space, $d \geq K$
    - $\mathbf{z} = \mathbf{W}^T\mathbf{x} \in R^l$
    - $\mathbf{W} \in R^{d\times l}, l \leq K-1$
    - $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_l]$
    - $\mathbf{z} = [z_1, \ldots, z_l]^T$
    - $z_i = \mathbf{w}_i^T\mathbf{x}, i = 1, \ldots, l$

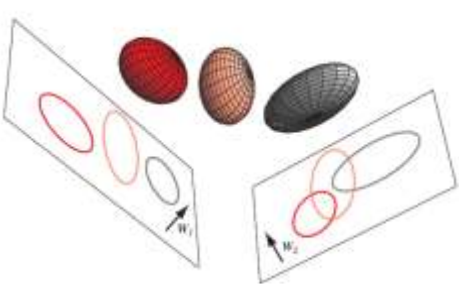


Fig. 3.6 [Duda 01]

Multiple Discriminant Analysis (2)

- MDA
  - $\mathbf{z} = \mathbf{W}^T\mathbf{x}$
  - $J_3(\mathbf{W}) = tr\left(\widetilde{\mathbf{S}_W}^{-1}\widetilde{\mathbf{S}_B}\right) = tr\left((\mathbf{W}^T\mathbf{S}_W\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{S}_B\mathbf{W})\right)$
    - $\frac{\partial}{\partial \mathbf{W}}J_3(\mathbf{W}) = \frac{\partial}{\partial \mathbf{W_1}}tr\left(\left(\mathbf{W_1}^T\mathbf{S}_W\mathbf{W}_1\right)^{-1}(\mathbf{W}^T\mathbf{S}_B\mathbf{W})\right)|_{\mathbf{W_1}=\mathbf{W}} +$
    $\frac{\partial}{\partial \mathbf{W_2}}tr\left((\mathbf{W}^T\mathbf{S}_W\mathbf{W})^{-1}(\mathbf{W_2}^T\mathbf{S}_B\mathbf{W}_2)\right)|_{\mathbf{W_2}=\mathbf{W}}$
    $= -2\mathbf{S}_W\mathbf{W}(\mathbf{W}^T\mathbf{S}_W\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{S}_B\mathbf{W})(\mathbf{W}^T\mathbf{S}_W\mathbf{W})^{-1} + 2\mathbf{S}_B\mathbf{W}(\mathbf{W}^T\mathbf{S}_W\mathbf{W})^{-1}$
    $= 0$
  - $\Longrightarrow \mathbf{S}_B\mathbf{W} = \mathbf{S}_W\mathbf{W}\widetilde{\mathbf{S}_W}^{-1}\widetilde{\mathbf{S}_B}$
  - $\Longrightarrow (\mathbf{S}_W^{-1}\mathbf{S}_B)\mathbf{W} = \mathbf{W}\left(\widetilde{\mathbf{S}_W}^{-1}\widetilde{\mathbf{S}_B}\right)$

$$\frac{\partial}{\partial \mathbf{A}}tr(\mathbf{AB}) = \frac{\partial}{\partial \mathbf{A}}tr(\mathbf{BA}) = \mathbf{B}^T$$
$$\frac{\partial}{\partial \mathbf{A}}tr(\mathbf{A}^T\mathbf{SA}) = (\mathbf{S}+\mathbf{S}^T)\mathbf{A}$$
$$\frac{\partial}{\partial \mathbf{A}}tr\left((\mathbf{A}^T\mathbf{SA})^{-1}\mathbf{B}\right) = -\mathbf{SA}(\mathbf{A}^T\mathbf{SA})^{-1}(\mathbf{B}+\mathbf{B}^T)(\mathbf{A}^T\mathbf{SA})^{-1}$$

Multiple Discriminant Analysis (3)

- MDA (cont.)
  - $(\mathbf{S}_W^{-1}\mathbf{S}_B)\mathbf{W} = \mathbf{W}\left(\widetilde{\mathbf{S}_W}^{-1}\widetilde{\mathbf{S}_B}\right)$
    - The two scatter matrices $\widetilde{\mathbf{S}_W}$, $\widetilde{\mathbf{S}_B}$ can be diagonalized simultaneously by a linear transformation
      - $\mathbf{B}^T\widetilde{\mathbf{S}_W}\mathbf{B} = \mathbf{I}$ and $\mathbf{B}^T\widetilde{\mathbf{S}_B}\mathbf{B} = \mathbf{D}$
        - » $\mathbf{I}$ and $\mathbf{D}$ are the within- and between-class scatter matrices of $\hat{\mathbf{z}}$
    - The transformed vector $\hat{\mathbf{z}}$
      - $\hat{\mathbf{z}} = \mathbf{B}^T\mathbf{z} = \mathbf{B}^T\mathbf{W}^T\mathbf{x} = \mathbf{C}^T\mathbf{x}$
        - » $\mathbf{C} = \mathbf{AB} \in R^{d\times l}$
      - No loss of the cost J_3
        - » $J_{3,\hat{\mathbf{z}}} = tr\left(\left(\mathbf{B}^T\widetilde{\mathbf{S}_W}\mathbf{B}\right)^{-1}(\mathbf{B}^T\widetilde{\mathbf{S}_B}\mathbf{B})\right) = \cdots = tr\left(\widetilde{\mathbf{S}_W}^{-1}\widetilde{\mathbf{S}_B}\right) = J_{3,\mathbf{z}}$
    - We finally obtain
      - $(\mathbf{S}_W^{-1}\mathbf{S}_B)\mathbf{C} = \mathbf{CD}$

A typical eigenvalue-eigenvector problem
Diagonal entries of $\mathbf{D}$: eigenvalues of $\mathbf{S}_W^{-1}\mathbf{S}_B$
Column of $\mathbf{C}$: the corresponding eigenvectors

Multiple Discriminant Analysis (4)

- MDA (cont.)
  - Note
    - The rank of $\mathbf{S}_B$ is at most $K-1$
    - The rank of $\mathbf{S}_W^{-1}\mathbf{S}_B$ is also at most $K-1$
      - There are at most $K-1$ nonzero eigenvalues
  - If $l = K-1$
    - First, form the matrix $\mathbf{C}$ so that its columns are the unit eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$
    - Then, the transformed vector is obtained by
      - $\hat{\mathbf{z}} = \mathbf{C}^T\mathbf{x}$
      - There is no loss in class separability power measured by J_3
        - » $J_{3,\mathbf{x}} = tr(\mathbf{S}_W^{-1}\mathbf{S}_B) = \lambda_1 + \lambda_2 + \cdots + \lambda_{K-1} + 0$
        - » $J_{3,\hat{\mathbf{z}}} = tr\big((\mathbf{C}^T\mathbf{S}_W\mathbf{C})^{-1}(\mathbf{C}^T\mathbf{S}_B\mathbf{C})\big) = tr\big((\mathbf{C}^T\mathbf{S}_W\mathbf{C})^{-1}(\mathbf{C}^T\mathbf{S}_W\mathbf{C}\mathbf{D})\big) = tr(\mathbf{D}) = \lambda_1 + \lambda_2 + \cdots + \lambda_{K-1} + 0$

The eigenvectors are not necessarily mutually orthogonal (because $\mathbf{S}_W^{-1}\mathbf{S}_B$ may not be symmetric)

Multiple Discriminant Analysis (5)

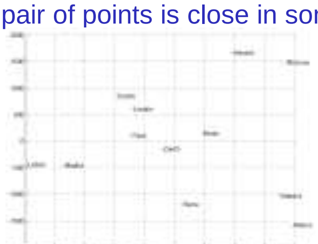
- MDA (cont.)
  - If $l < K-1$
    - The matrix $\mathbf{C}$ is formed the eigenvectors corresponding to the l largest eigenvalues of $\mathbf{S}_W^{-1}\mathbf{S}_B$
      - There is loss in class separability power
      - $J_{3,\hat{\mathbf{z}}} < J_{3,\mathbf{x}}$
  - If $\mathbf{S}_W$ is not invertible (when $N < d$)
    - Use the pseudoinverse $\mathbf{S}_W^{+}$, or
    - Use regularization techniques
      - $\mathbf{S}_W \longleftarrow \mathbf{S}_W + \sigma\mathbf{\Omega}$
        - » $\mathbf{\Omega}$ can be any positive definite and symmetric matrix
    - Or, adopt a two-stage approach
      - e.g. PCA+LDA

Fisherfaces

- Fisherfaces
  - Using LDA to reduce the dimension to at most $K-1$
- In the face recognition problem
  - The within-class scatter matrix is usually singular
    - $\mathbf{S}_W = \sum_{i=1}^{K} P_i \mathbf{S}_i = \frac{1}{N}\sum_{i=1}^{K}\sum_{\mathbf{x}\in C_i}(\mathbf{x}-\boldsymbol{\mu}_i)(\mathbf{x}-\boldsymbol{\mu}_i)^T \in R^{d\times d}$
    - Because its rank is at most $(N-K) \ll d$
  - To avoid the problem
    - 1) Use PCA to reduce the dimension to $(N-K)$
    - 2) Apply LDA to reduce the dimension to $(K-1)$

Multidimensional Scaling (1)

- Multidimensional scaling (MDS)
  - Given an $N\times N$ matrix of dissimilarities for centered data
    - $\mathbf{D} = \begin{bmatrix} d(\mathbf{x}_1,\mathbf{x}_1) & \cdots & d(\mathbf{x}_1,\mathbf{x}_N) \\ d(\mathbf{x}_2,\mathbf{x}_1) & & d(\mathbf{x}_2,\mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ d(\mathbf{x}_N,\mathbf{x}_1) & \cdots & d(\mathbf{x}_N,\mathbf{x}_N) \end{bmatrix}$
  - Find a *configuration* $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ so that the distance between a pair of points is close in some sense to the dissimilarities



Map of Europe drawn by MDS

Given road travel distances between cities
Use MDS to get an approximation to the map

Fig. 6.9 [Alpaydin, 2014]

Multidimensional Scaling (2)

- Classical MDS
  - Given an $N \times N$ dissimilarity matrix $\mathbf{D}$
    - Assuming the dissimilarities are Euclidean distance
  - To determine the coordinates of a set of points
    - $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}, \mathbf{x}_i \in R^d, \sum_{i=1}^{N} \mathbf{x}_i = \mathbf{0}$
  - Let $\mathbf{K} = \mathbf{X}\mathbf{X}^T$
    - $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$ is an $N \times d$ matrix
    - Thus
      - $K(i,j) = \mathbf{x}_i^T \mathbf{x}_j$
        - » The dot product of $\mathbf{x}_i$ and $\mathbf{x}_j$
      - $\mathbf{K}$ is a matrix of pairwise similarity
    - The distance between two data points are
      - $D(i,j) = d_{ij}^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2 = K(i,i) + K(j,j) - 2K(i,j)$

Multidimensional Scaling (3)

- Classical MDS (cont.)
  - Given the dissimilarity matrix $\mathbf{D}$
  - Construct the $N \times N$ symmetric matrix $\mathbf{K}$
    - $K(i,j) = -\frac{1}{2}(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2)$
      - $d_{i.}^2 = \frac{1}{N}\sum_{j=1}^{N} d_{ij}^2,\ d_{.j}^2 = \frac{1}{N}\sum_{i=1}^{N} d_{ij}^2,\ d_{..}^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^2$
  - Factorize $\mathbf{K}$
    - $\mathbf{K} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$
      - $\mathbf{u}_i, \lambda_i$: the eigenvector and eigenvalue of $\mathbf{K}$
  - The matrix of coordinates
    - $\mathbf{X} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}$
  - The reduced-dimension representation
    - $\hat{\mathbf{X}}_{N\times l} = \mathbf{U}_l \boldsymbol{\Lambda}_l^{1/2}$
      - If we start with a set of data (rather than a distance matrix), then the reduced-dimension representation is the same as carrying out PCA

Multidimensional Scaling (4)

- Metric multidimensional scaling
  - Given an $N \times N$ matrix of dissimilarities for centered data $\mathbf{x}_i$
  - Find a configuration $\mathbf{y}_i$ in a dimension l
    - So that the distance between a pair of points in that space correspond to the distance between points in the original space

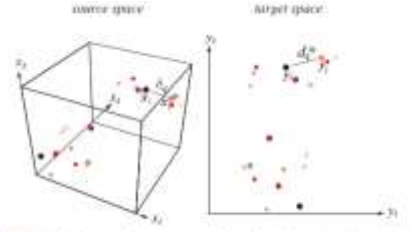


FIGURE 10.26. The figure shows an example of points in a three-dimensional space being mapped to a two-dimensional space. The size and color of each point $\mathbf{x}_i$ matches that of its image, $\mathbf{y}_i$. Here we use simple Euclidean distance, that is, $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ and $d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$. In typical applications, the source space usually has high dimensionality, but to allow easy visualization the target space is only two- or three-dimensional. From:

Fig. 10.26 [Duda 01]

Multidimensional Scaling (5)

- Metric multidimensional scaling (cont.)
  - Assume
    - δ_{ij}: distance between $\mathbf{x}_i$ and $\mathbf{x}_j$
    - d_{ij}: distance between $\mathbf{z}_i$ and $\mathbf{z}_j$
  - The criterion functions
    - $J_{ee} = \frac{\sum_{i<j}(d_{ij}-\delta_{ij})^2}{\sum_{i<j}\delta_{ij}^2}$ $\quad J_{ff} = \sum_{i<j}\left(\frac{d_{ij}-\delta_{ij}}{\delta_{ij}}\right)^2$
    - $J_{ef} = \frac{1}{\sum_{i<j}\delta_{ij}}\sum_{i<j}\frac{(d_{ij}-\delta_{ij})^2}{\delta_{ij}}$
  - The algorithm: standard gradient-descent procedure
    - Starting with some initial configuration $\mathbf{z}_i, i = 1, \ldots, N$
    - Changing $\mathbf{z}_i, i = 1, \ldots, N$ in the direction of greatest rate of decrease in the criterion function

Multidimensional Scaling (6)

- Metric multidimensional scaling (cont.)
  - Assume we use Euclidean distance $d_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|$
  - Then the gradients are easy to compute
    - $\nabla_{\mathbf{z}_k} J_{ee} = \frac{2}{\sum_{i<j} \delta_{ij}^2} \sum_{j \neq k} \frac{(d_{kj} - \delta_{kj})(\mathbf{z}_k - \mathbf{z}_j)}{d_{kj}}$
    - $\nabla_{\mathbf{z}_k} J_{ff} = 2 \sum_{j \neq k} \frac{(d_{kj} - \delta_{kj})(\mathbf{z}_k - \mathbf{z}_j)}{\delta_{kj}^2 d_{kj}}$
    - $\nabla_{\mathbf{z}_k} J_{ef} = \frac{2}{\sum_{i<j} \delta_{ij}} \sum_{j \neq k} \frac{(d_{kj} - \delta_{kj})(\mathbf{z}_k - \mathbf{z}_j)}{\delta_{kj} d_{kj}}$

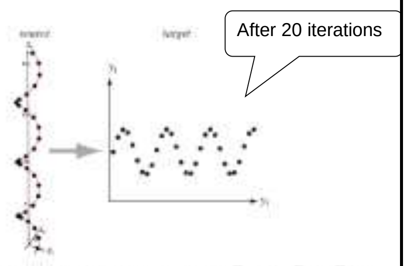


FIGURE 10.27. Thirty points of the form $\mathbf{x} = (\cos(k/\sqrt{2}), \sin(k/\sqrt{2}), k/\sqrt{2})^t$ for $k = 0, 1, \ldots, 29$ are shown at the left. Multidimensional scaling using the J_{ef} criterion (Eq. 109) and a two-dimensional target space leads to the image points shown at the right. This lower-dimensional representation shows clearly the fundamental sequential nature of the points in the original source space. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Fig. 10.26 [Duda 01]

Laplacian Eigenmaps (1)

- Goal
  - To compute the low dimension representation of the data that optimally preserves local neighborhood information
    - Given $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \mathbf{x}_i \in R^d$
    - Find a set $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}, \mathbf{z}_i \in R^l$ $(l \ll d)$ such that $\mathbf{z}_i$ represents $\mathbf{x}_i$
- Step 1 (constructing the adjacency graph)
  - Construct a weighted graph G with N nodes, one for each point $\mathbf{x}_i$
    - Two vertices are connected with an edge if
      - $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon$ or
      - One is among the k-nearest neighbors of the other
- Step 2 (choosing the weights)
  - If $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected, $W(i,j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2)$
  - Otherwise $W(i,j) = 0$

Laplacian Eigenmaps (2)

- Step 3 (eigenmaps)
  - Assume the G is connected
    - Otherwise proceed with step 3 for each connected component
      - Let $\mathbf{W}$ be the $N \times N$ symmetric weight matrix for the graph
  - Let $\mathbf{D}$ be the diagonal weight matrix $D_{ii} = \sum_j W(i,j)$
    - Its entries are column (or row) sums of $\mathbf{W}$
  - Define the graph Laplacian matrix $\mathbf{L}$ by $\mathbf{L} \equiv \mathbf{D} - \mathbf{W}$
    - Is symmetric and positive semidefinite
  - Perform the generalized eigendecomposition $\mathbf{Lv} = \lambda \mathbf{Dv}$
  - Let $0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_l$ be the smallest $l+1$ eigenvalues
  - Ignore the eigenvector corresponding to $\lambda_0 = 0$
  - Choose the next l eigenvectors $\mathbf{v}_1, \mathbf{v}, \dots, \mathbf{v}_l$
  - Map $\mathbf{x}_i \in R^d \mapsto \mathbf{z}_i = [\mathbf{v}_1(i), \mathbf{v}_2(i), \dots, \mathbf{v}_l(i)]^T \in R^l, i = 1, \dots, N$

Laplacian Eigenmaps (3)

- Justification (for the case of $l = 1$)
  - To map $\mathbf{x}_i \mapsto z_i = [\mathbf{v}_1(i)], i = 1, \dots, N$ so that the neighbors stay as close as possible after the mapping
  - The criterion
    - $E_L = \sum_{i=1}^N \sum_{j=1}^N (z_i - z_j)^2 W(i,j)$

      $= \sum_{i=1}^N \sum_{j=1}^N (z_i^2 + z_j^2 - 2z_i z_j)^2 W(i,j)$

      $= \sum_{i=1}^N z_i^2 \sum_{j=1}^N W(i,j) + \sum_{j=1}^N z_j^2 \sum_{i=1}^N W(i,j) - 2\sum_{i=1}^N \sum_{j=1}^N z_i z_j W(i,j)$

      $= \sum_{i=1}^N z_i^2 D_{ii} + \sum_{j=1}^N z_j^2 D_{jj} - 2\sum_{i=1}^N \sum_{j=1}^N z_i z_j W(i,j)$

    - $2\mathbf{z}^T\mathbf{Lz} \ge 0$
      - $\mathbf{z} = [z_1, \dots, z_N]^T$

For larger similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, the distance between z_i and z_j should be smaller

  - The minimization problem
    - $\underset{\mathbf{z}^T\mathbf{Dz}=1}{\operatorname{argmin}} \mathbf{z}^T\mathbf{Lz}$

Laplacian Eigenmaps (4)

- Example [Belkin and Niyogi, 2002]

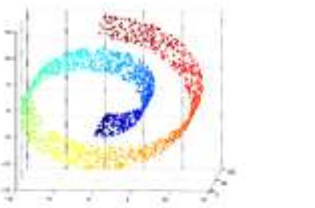
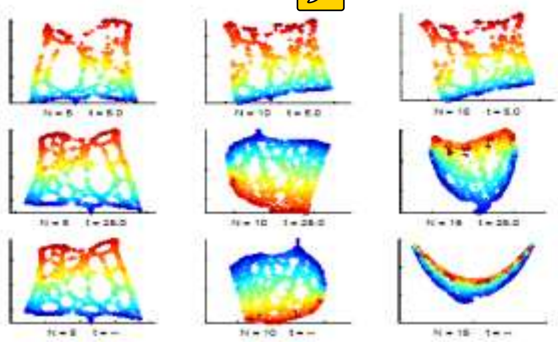


Figure 1: 2000 random data points on the "swiss roll"



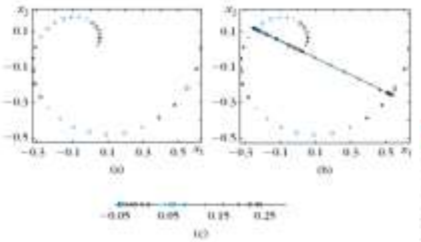
$$W(i,j) = \frac{\exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)}{t}$$

Figure 2: Two-dimensional representations of the "swiss roll" data, for different values of the number of nearest neighbors N and the heat kernel parameter t. $t = \infty$ corresponds to the discrete weights.

[Belkin and Niyogi, 2002]

Laplacian Eigenmaps (5)

- Example 6.6 (p. 361 [Theodoridis et. al.])
  - The data lies on a nonlinear manifold
  - PCA
    - Neighboring information is lost
  - Laplacian eigenmap
    - The points are unfolded in an 1-D straight line
    - Neighboring information is retained in this 1-D representation



PCA:
$\lambda_0 = 0.089, \lambda_1 = 0.049$

Laplacian eigenmap
$\varepsilon = 0.2$
$\sigma = \sqrt{0.5}$

FIGURE 6.6
(a) A spiral of Archimedes in the two-dimensional space. (b) The previous spiral together with the projections of the sampled points on the direction of the first principal component, resulting from PCA. It is readily seen that neighboring information is lost after the projection and points corresponding to different parts of the spiral overlap. (c) The one-dimensional map of the spiral using the Laplacian method. In this case, the neighboring information is retained after the nonlinear projection and the spiral nicely unfold to a one-dimensional line.

Laplacian Eigenmaps (6)

- Example 6.7 (p. 362 [Theodoridis et. al.])
  - A 3-D spiral data 2-D mapping using the Laplacian eigenmap

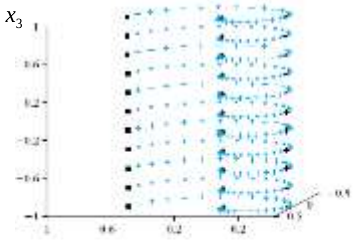
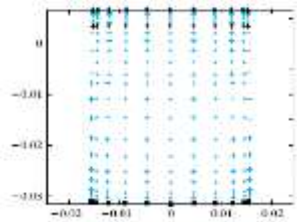


FIGURE 6.7
Samples from a three-dimensional spiral. One can think of it as a number of two-dimensional spirals one above the other. Different symbols have been used in order to track neighboring information.



$\varepsilon = 0.2$
$\sigma = \sqrt{0.5}$

All points corresponding to the same x_3 are mapped across the same line

FIGURE 6.8
Two-dimensional mapping of the spiral of Figure 6.7 using the Laplacian eigenmap method. The three-dimensional structure is unfolded to the two-dimensional space by retaining the neighboring information.

Locally Linear Embedding (1)

- Locally linear embedding (LLE)
  - Modeling the manifold as a union of locally linear patches
    - Express each $\mathbf{x}_i$ as a linear combination of its neighbors
    - Construct $\mathbf{z}_i$ so that they can be expressed as *the same linear combination* of their corresponding neighbors

[Roweis and Saul, 2000]

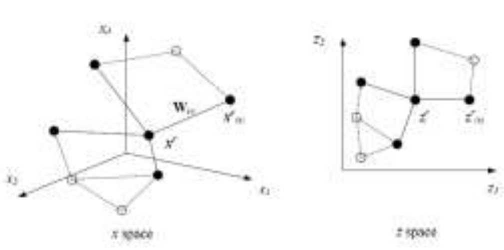
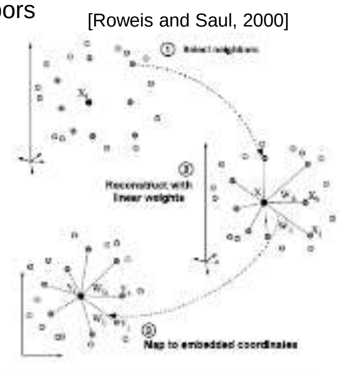


Fig. 6.15 [Alpaydin]



Locally Linear Embedding (2)

- Step 1
  - For each point $\mathbf{x}_i$, find its k nearest neighbors
- Step 2
  - Compute the weights $W(i,j)$ by minimizing
    - $E_W = \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j=1}^{N} W(i,j)\mathbf{x}_{i_j} \right\|^2$
      - $\mathbf{x}_{i_j}$ is the neighbor of $\mathbf{x}_i$
      - $\sum_{j=1}^{N} W(i,j) = 1$
  - The Largrangian
    - $L = \sum_{i=1}^{N} F_i$

$$= \sum_{i=1}^{N} \left(\frac{1}{2} \left\| \mathbf{x}_i - \sum_{j\in N(i)} W(i,j)\mathbf{x}_{i_j} \right\|^2 - \lambda_i \left(\sum_{j\in N(i)} W(i,j) - 1 \right) \right)$$

Locally Linear Embedding (3)

- Step 2 (cont.)
  - Each F_i can be minimized separately
    - $F = \frac{1}{2} \left\| \mathbf{x} - \sum_j W_j \mathbf{x}_j \right\|^2 - \lambda \left(\sum_j W_j - 1 \right)$
  - Define the neighborhood correlation matrix $\mathbf{C}$ and the vector $\mathbf{b}$
    - $C_{jk} = \langle \mathbf{x}_j, \mathbf{x}_k \rangle$
    - $\mathbf{b} = \langle \mathbf{x}, \mathbf{x}_j \rangle$
  - $\forall k, \frac{\partial F}{\partial W_k} = 0$
    - $\lambda \mathbf{e} = \mathbf{C}\mathbf{w} - \mathbf{b}$
    - $\mathbf{w} = \mathbf{C}^{-1}(\lambda \mathbf{e} + \mathbf{b})$
      - $\mathbf{e}$ is the vector of all ones
      - $\lambda = \frac{1 - \mathbf{e}^T \mathbf{C}^{-1} \mathbf{b}}{\mathbf{e}^T \mathbf{C}^{-1} \mathbf{e}}$

$\because \mathbf{e}^T \mathbf{w} = 1$
$\therefore \lambda \mathbf{e}^T \mathbf{C}^{-1} \mathbf{e} + \mathbf{e}^T \mathbf{C}^{-1} \mathbf{b} = 1$

Locally Linear Embedding (4)

- Step 3
  - Given $\mathbf{W}$, find $\{\mathbf{z}_1,\mathbf{z}_2,\dots,\mathbf{z}_N\}, \mathbf{z}_i \in R^l \ (l \ll d)$ by minimizing
    - $E_z = \sum_{i=1}^{N} \left\| \mathbf{z}_i - \sum_{j=1}^{N} W(i,j)\mathbf{z}_j \right\|^2$
      - By requiring $\mathbf{z}$ to be zero mean and unit covariance
  - The Largrangian
    - $L = \sum_{i=1}^{N} \left\| \mathbf{z}_i - \sum_{j=1}^{N} W(i,j)\mathbf{z}_j \right\|^2 - \sum_{a,b} \lambda_{a,b} \left(\sum_{i=1}^{N} \frac{1}{N} Z_{i,a} Z_{i,b} - \delta_{a,b} \right)$
      - where $\mathbf{Z} = [\mathbf{z}_1,\mathbf{z}_2,\dots,\mathbf{z}_N]^T$
    - $\frac{\partial L}{\partial Z_{k.l}} = 0 \Rightarrow \cdots \Rightarrow (\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\mathbf{Y} = \frac{1}{N}\mathbf{Y}\boldsymbol{\Lambda}$
    - Performing the eigendecomposition of $(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$
    - Discarding the eigenvector that corresponds to the smallest eigenvalue
    - Taking the next (lower) eigenvectors to form the low-dimensional outputs

Isomap (1)

- Isometric mapping
  - Motivation
    - Geodesic distances can better reflect the true low-dimensional geometry than Euclidean distances

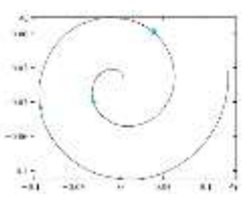
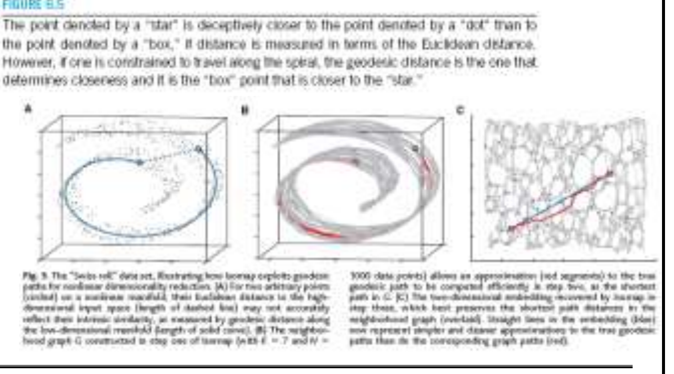


Fig. 6.5



[Tenenbaum et. al, 2000]

Isomap (2)

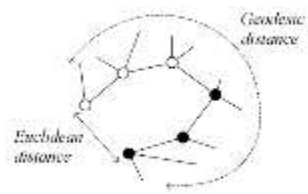


Fig. 6.14 [Alpaydin]

- Step 1
  - Constructing the adjacency graph
    - The edge weight $W(i,j) \propto \|\mathbf{x}_i - \mathbf{x}_j\|^2$
- Step 2
  - Estimating the pairwise geodesic distances among all pairs of points by computing their shortest path distance in the graph
    - e.g. Dijkstra's algorithm
- Step 3 (classical MDS)
  - Performing the eigendecomposition of the Gram matrix
  - Selecting the l most significant eigenvectors to represent the low-dimensional space
    - $\mathbf{z}_i = \left[\sqrt{\lambda_1}\mathbf{u}_1(i), \dots, \sqrt{\lambda_l}\mathbf{u}_l(i)\right]^T$

Locality Preserving Projection (1)

- The nonlinear dimension reduction techniques are defined on the training data
  - Unclear for mapping new test data
    - The new point should be added to the dataset
    - The algorithm needs to be run once more using $N+1$ instances
- Locality preserving projection (LPP)
  - A linear approximation of the nonlinear Laplacian eigenmap
  - Goal
    - Given $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}, \mathbf{x}_i \in R^d$
    - Find a transformation matrix $\mathbf{A}$
      - $\mathbf{z}_i = \mathbf{A}^T\mathbf{x}_i \in R^l$
      - $\mathbf{A} \in R^{d \times l}$
    - Such that $\mathbf{z}_i$ represents $\mathbf{x}_i$

Locality Preserving Projection (2)

- Step 1 (constructing the adjacency graph)
- Step 2 (choosing the weights)
- Step 3 (eigenmaps)
  - Perform the generalized eigendecomposition
    - $\mathbf{XLX}^T\mathbf{a} = \lambda\mathbf{XDX}^T\mathbf{a}$
      - $D_{ii} = \sum_j W(i,j)$
      - $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix
  - Let $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_l$ be the solutions, ordered according to $\lambda_1 < \cdots < \lambda_l$
    - $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_l]$
  - Then
    - $\mathbf{x}_i \in R^d \mapsto \mathbf{z}_i = \mathbf{A}^T\mathbf{x}_i, i = 1, \dots, N$

Locality Preserving Projection (3)

- Justification (for the case of $l = 1$)
  - To map $\mathbf{x}_i \in R^d \mapsto z_i = \mathbf{a}^T\mathbf{x}_i, i = 1, \dots, N$ so that the neighbors stay as close as possible after the mapping
  - The criterion
    - $E_L = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(z_i - z_j\right)^2 W(i,j)$

      $= \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j\right)^2 W(i,j)$

      $= \sum_{i=1}^{N}\mathbf{a}^T\mathbf{x}_i D_{ii}\mathbf{x}_i^T\mathbf{a} - \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{a}^T\mathbf{x}_i W(i,j)\mathbf{x}_j^T\mathbf{a}$

      $= \mathbf{a}^T\mathbf{X}(\mathbf{D} - \mathbf{W})\mathbf{X}^T\mathbf{a}$
    - The constraint
      - $\mathbf{z}^T\mathbf{D}\mathbf{z} = 1 \Rightarrow \mathbf{a}^T\mathbf{X}\boldsymbol{D}\mathbf{X}^T\mathbf{a} = 1$
  - The minimization problem
    - $\underset{\mathbf{a}^T\mathbf{X}\boldsymbol{D}\mathbf{X}^T\mathbf{a}=1}{\operatorname{argmin}} \;\; \mathbf{a}^T\mathbf{X}\boldsymbol{L}\mathbf{X}^T\mathbf{a} \;\; = 1$